(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 932 109 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
28.07.1999 Bulletin 1999/30

(51) Int Cl.[6]: **G06F 17/30**

(21) Application number: **99400130.3**

(22) Date of filing: **21.01.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **22.01.1998 US 10571**

(71) Applicant: **YEDA RESEARCH & DEVELOPMENT COMPANY, LTD.**
76100 Rehovot (IL)

(72) Inventors:
- **Naor, Moni**
  Tel Aviv 69122 (IL)
- **Nissim, Yaacov**
  Ramat-Gan 52525 (IL)

(74) Representative: **Moutard, Pascal Jean et al**
Cabinet Beau de Loménie
158, rue de l'Université
F-75340 Paris Cedex 07 (FR)

(54) A method for authentification item

(57) A memory containing an authenticated search tree that serves for authenticating membership or non membership of items in a set. The authenticated search tree including a search tree having nodes and leaves and being associated with a search scheme. The nodes including dynamic search values and the leaves including items of the set The nodes are associated, each, with a cryptographic hash function value that is produced by applying a cryptographic hash function to the cryptographic hash values of the children nodes and to the dynamic search value of the node. The root node of the authenticated search tree is authenticated by a digital signature.

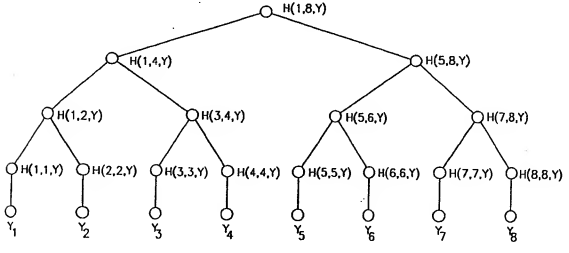

FIG.1

EP 0 932 109 A2

Description

FIELD OF THE INVENTION

[0001] The present invention is in the general field of digital signature for authentication purposes.

BACKGROUND OF THE INVENTION

[0002] The wide use of public key cryptography requires the ability to verify the authenticity of public keys. This is achieved through the use of certificates (that serve as a mean for transferring trust) in a *Public Key Infrastructure* (PKI). A certificate is a message signed by a publicly trusted authority (the certification authority, whose public key authenticity may be provided by other means) which includes a public key and additional data, such as expiration date, serial number and information regarding the key and the subject entity.

[0003] When a certificate is issued, its validity is limited by an expiration date. However, there are circumstances (such as when a private key is revealed, or when a key holder changes affiliation or position) where a certificate must be revoked prior to its expiration date. Thus, the existence of a certificate is a necessary but not sufficient evidence for its validity, and a mechanism for determining whether a certificate was revoked is needed.

[0004] A typical application is a credit card system where the credit company may revoke a credit card, temporarily or permanently, prior to its expiration, e.g. when a card is reported stolen or according to its user's bank account balance.

PRIOR ART DISCUSSION:

Certificate Revocation List (CRL)

[0005] A CRL is a signed list issued by the CA identifying all revoked certificates by their serial numbers. The list is concatenated with a time stamp (as an indication of its freshness) and signed by the CA that originally issued the certificates. The CRLs are sent to the directory on a periodic basis, even if there are no changes, to prevent the malicious replay of old CRLs instead of new CRLs.

[0006] As an answer to a query, the directory supplies the most updated CRL (the complete CRL is sent to the merchant).

- The main advantage ofthe scheme is its simplicity.
- The main disadvantage of the scheme is its high directory-to-user communication costs (since CRLs may get very long). Another disadvantage is that a user may not hold a succinct proof for the validity of his certificate.

[0007] A reasonable validity expiration period should be chosen for certificates. If the expiration period is short, resources are wasted reissuing certificates. If the expiration period is long, the CRL may get long, causing high communication costs and difficulties in CRL management Kaufman *et al.* [15, Section 7.7.3] suggested reissuing all certificates whenever the CRL grows beyond some limit. In their proposal, certificates are marked by a serial number instead of an expiration date. (Serial numbers are incremented for each issued certificate. Serial numbers are not reused even when all certificates are reissued.) The CRL contains a field indicating the *first valid* certificate. When all certificates are reissued, the CRL first valid certificate field is updated to contain the serial number of the first reissued certificate.

Certificate Revocation System

[0008] Micali [18) suggested the Certificate Revocation system (CRS) in order to improve the CRL communication costs. The underlying idea is to sign a message for every certificate stating whether it was revoked or not, and to use an off-line/on-line signature scheme [11] to reduce the cost of periodically updating these signatures.

[0009] To create a certificate, the CA associates with each certificate two numbers (Y_{365} and N) that are signed along with the 'traditional' certificate data. For each certificate, the CA chooses (pseudo) randomly two numbers N_0 Y_0 and computes (using a one-way function f) $Y_{365} = f^{365}(Y_0)$ and $N = f(N_0)$. (Actually, a stronger assumption on f is required, e.g. that f is one-way on its iterates, i.e. that given $y = f(x)$ it is infeasible to find x' such that $y = f(x')$. This is automatically guaranteed if f is a one-way permutation.)

[0010] The directory is updated daily by the CA sending it a number C for each certificate as follows:

1. For a non-revoked certificate, the CA reveals one application of f, i.e. $C = Y_{365-i} = f^{365-i}(Y_0)$, where i is a daily

incremented counter, $i = 0$ on the date of issue.

2. For a revoked certificate, $C = N_0$.

[0011] Thus the most updated value for C serves as a short proof (that certificate x was or was not revoked) that the directory may present in reply to a user query x.

- The advantage of CRS over CRL is in its query communication costs. Based on Federal PKI (Public Key Infrastructure) estimates, Micali [18] showed that although the daily update of the CRS is more expensive than a CRL update, the cost of CRS querying is much lower. He estimated the resulting in 900 fold improvement in total communication costs over CRLs.

[0012] Another advantage of CRS is that each user may hold a succinct transferable proof ofthe validity of his certificate. Directory accesses are saved when users hold such proofs and presents them along with their certificates.

- The main disadvantage of this system is the increase in the CA-to-directory communication (It is of the same magnitude as directory-to-users communication, where the existence of a directory is supposed to decrease the CA's communication). Moreover, since the CA's communication costs are proportional to the directory update rate, CA-to-directory communication costs limit the directory update rate.

[0013] The complexity of verifying that a certificate was not revoked is also proportional to the update rate. For example, for an update once an hour, a user may have to apply the function, f, 365 x 24 = 8760 times in order to verify that a certificate was not revoked, making it the dominant factor in verification.

Certificate Revocation Trees

[0014] Kocher [16] suggested the use of Certificate Revocation Trees (CRT) referred to also as authentication tree, in order to enable the verifier of a certificate to get a short proof that the certificate was not revoked. A CRT is a hash tree with leaves corresponding to a set of statements about certificate serial number X issued by a CA, CA_X. The set of statements is produced from the set of revoked certificates of every CA. It provides the information whether a certificate X is revoked or not (or whether its status is unknown to the CRT issuer). There are two types of statements: specifying ranges of unknown CAs, and, specifying certificates range of which only the lower certificate is revoked. For instance, if CA_1 revoked two certificates, $X1 < X2$, than one of the statements is:

$$\text{if } CA_x = CA_1 \text{ and } X_1 \le X < X_2 \text{ then } X \text{ is revoked if } X = v$$

[0015] To produce the CRT, the CRT issuer builds a binary hash tree **[17]** with leaves corresponding to the above statements

[0016] A proof for a certificate status is a pah in the hash tree, from the root to the appropriate leaf (statement) specifying for each node on the path the values of its children.

- The main advantages of CRT over CRL are that the entire CRL is not needed for verifying a specific certificate and that a user may hold a succinct proof of the validity of his certificate.

- The main disadvantage of CRT is in the computational work needed to update the CRT. Any change in the set of revoked certificates may result in re-computation of the *entire* CRT.

LIST OF CITED PRIOR ART

U.S. patent 4,309,569 (hereinafter the *Merkle* patent)

[0017]

[1] A. V. Aho, J. E. Hopcroft, J. D. Ullman. "Data Structures and Algorithms". Addison-Wesley, 1983.

[2] R.G. Seidel , C.R Aragon "Randomized Search Trees". Proc. 30th Annual IEEE Symp. on Foundations of Computer Science, pp. 540-545,

1989.

[6] M. Bellare, P.Rogaway. "Collision-Resistant Hashing: Towards Making UOWHFs Practical". Advances in Cryptology - CRYPTO '97, Lecture Notes in Computer Science, Springer-Verlag, 1997.

[11] S. Even, O. Goldreich, S. Micali. "On-Line/Off-Line Digital Signatures". Journal of Cryptology, Springer-Verlag, Vol. 9 pp. 35-67, 1996.

[12] O. Goldreich, S. Goldwasser, and S. Halevi,. "Collision-Free Hashing from Lattice Problems". *ECCC,* TR96-042, 1996.
http: //www. eccc. unitrier. de/eccc/

[13] A. Herzberg, H. Yochai. "Mini-Pay: Charging per Click on the Web". Proc. 6th International World Wide Web Conference, 1997.
http: //www6. nttlabs. com/

[15] C. Kaufman, R. Perlman, M. Speciner. "Network Security. Private Communiation in a Public World". Prentice Hall series in networking and distributing systems, 1995.

[16] P. Koeher. "A Quick Introduction to Certificate Revocation Trees (CRTs)".
http: //www. valicert. com/company/crt. html

[17] R. C. Merkle. "A Certified Digital Signature". Proc. Crypto '89, Lecture Notes in Computer Science 435, pp. 234-246, Springer-Verlag, 1989.

[18] S. Micali. "Efficient Certificate revocation". Technical Memo MIT/LCS/TM542b, 1996.

GLOSSARY

[0018] There follows glossary of terms some of which conventional and others have been coined:

[0019] ***Certification Authority (CA)*** - A trusted party, already having a certified public key, responsible for establishing and vouching for the authenticity of public keys and/or other information such as credit card numbers.

[0020] A CA preferably, but not necessarily, does not provide on-line certificate information services to users. Instead, it updates a directory on a periodic basis). As will be shown below in some embodiments directories are not used.

[0021] A CA issues certificates for users by a message containing the certificate serial number relevant data and an expiration date. The certificate is sent to a directory and/or given to the user. The CA may revoke a certificate prior to its expiration date. *Certificate* is by no means bound to the latter definition and may encompass data pertain to e.g. one or more (such as range of) public key(s), credit card number(s), and others; presented either in explicit form or after having been subject to a function such as encoding or encryption. (the term *item* and *certificate* are used in the specification interchangeably)

[0022] ***Directory - :*** One or more non-trusted parties that get updated certificate revocation information from the CA and serve as a certificate database accessible by the users.

[0023] ***User*** - A non-trusted party that receives its certificate from the CA and issues queries for certificate information. User should be construed as encompassing among others:

(i) a merchant who queries the validity of other users' certificates,
(ii) a user who gets proof of the validity of his/her certificate for using it *vise-vis* other users.

[0024] ***Search free*** - A well known data structure that is associated with search scheme which enables to construct a search path in the tree, from the root to a sought item (associated with a leaf). The search path exploits search values that reside in the tree nodes and possibly also in the links. Search tree is inherently designed to handle update transactions (i.e. delete and/or insert items to the tree). Typical, yet not exclusive, examples of search trees being: 2-3 tree, Btree, Btree+, TriS, treaps and others.

[0025] ***Update Transactions*** - Insert new item to a tree; delete existing item in a tree.

[0026] ***Authentication Tree*** - a rooted tree where each internal node authenticates the values of its children by

means of a cryptographic hash function and the root is authenticated by means of a digital signature. Typical, yet not exclusive, example is illustrated in the *Merkle* patent

[0027] ***Cryptographic hash function-*** includes:

(i) *collision intractable function* h() such that it is computationally essentially infeasible to find y≠x satisfying h(x) =h(y). Typical, yet not exclusive example is illustrated in the *Merkle* patent; or
(ii) *universal one way hash function* h() such that there exists a family of functions h() such that for every x and random h() from the family, it is computationally essentially infeasible to find y≠x satisfying h(x)=h(y). (for detailed discussion in (I) and (II), see [6]).

SUMMARY OF THE INVENTION

[0028] There is, accordingly, a need in the art for eliminating or substantially reducing the drawbacks associated with hitherto known techniques by providing a novel technique for authenticating items.

[0029] The present invention incorporates the utilization of conventional authentication trees as well as conventional search trees such as 2-3 tree or Btree. The utilization of search trees enables to authenticate an item (or items) whilst obviating the need to transmit a large amount of data to this end. The utilization of the authentication tree, according to the prior art, enables to transmit a series of revoked, (or otherwise) valid items.

[0030] The major drawback of using an authentication tree, e.g. of the kind disclosed in the *Merkle* patent, arises when the latter is subject to modification transactions. The latter bring about new arrangement of items in the leaves and, consequently, (as will be exemplified below), necessitates the modification of the values of multitude nodes (hereinafter modified nodes) in the tree.

[0031] Not only is an extensive computation required in order to update the values of the modified nodes, but also by utilizing an authentication high communication overhead is imposed when the multitude values of said modified nodes are transmitted over the communication network, e.g. from the CA to the directory. Considering that such modification may occur quiet frequently, the specified overhead renders the use of prior art authentication trees commercially infeasible.

[0032] According to the invention, a conventional authentication tree is "superimposed" on conventional search tree (bringing about authentication search tree) benefiting thus both from the inherent advantages of the authentication tree insofar as authenticating items is concerned and from the limited changes that are imposed on the tree nodes due to the search tree structure.

[0033] Accordingly, the present invention provides for a memory containing an authenticated search tree that serves for authenticating membership or non membership of items in a set; the authenticated search tree, comprising:

a search tree having nodes and leaves and having associated therewith a search scheme; the nodes including dynamic search values and the leaves including items of said set; the nodes are associated, each, with a cryptographic hash function value that is produced by applying a cryptographic hash function to at least: (I) the cryptographic hash values of the children nodes and (II) the dynamic search value of said node;
at least the root node of said authenticated search tree is authenticated by a digital signature.

[0034] Still further the invention provides for a method for authenticating membership or non membership of items in a set; comprising:

(i) providing an authenticated search tree of the kind specified;
(ii) authenticating at least one item of said set by computing the authentication path as induced by said at least one item and the root.

[0035] Still further the invention provides for a method for updating at least one item of a set in an authenticated search tree, comprising:

(i) providing a search authenticated tree of the kind specified;;
(ii) updating said search tree so as to obtain updated nodes;
(iii) computing an authentication path as induced by said updated nodes; and
(iv) authenticating at least said root modified node by a digital signature.

[0036] It should be noted that the specified order does not necessarily imply that in iterative procedure all the steps are performed in each iteration. Thus for example the steps (ii) and (iii) may be performed in each iteration and step (iv) may be applied once at the last iteration. This, likewise, applies to the other aspects of method and system as

described herein.

[0037] The invention further provides a system for authenticating/updating *mutatis mutandis*.

BRIEF DESCRIPTION OF THE DRAWINGS

[0038] The invention will now be described, by way of example only with reference to the accompanying drawings, in which:

Fig. 1 illustrates an authentication tree according to the prior art;
Figs. 2A-B illustrate a search authenticated tree according to one embodiment of the invention;
Fig. 3 illustrates a system configuration according to one embodiment of the invention;
Fig. 4 illustrates a system configuration according to another embodiment of the invention; and
Fig. 5A-B illustrate a manner in which a search authenticated tree is updated according to the embodiment of Fig. 4.

DESCRIPTION OF SPECIFIC EMBODIMENTS

[0039] Attention is first directed to Fig. 1 illustrating an authentication tree according to the prior art e.g as disclosed in the specified *Merkle* patent, the contents of which are incorporated herein by reference.

[0040] Consider, for example, that certificates Y_1 to Y_8 stand for a certificate list (CL) of all valid credit cards. Now, a user wants to use his credit card Y_5 in a commercial transaction *vis-a-vis* a merchant. The merchant addresses a directory that holds the authentication tree (i.e. authentication tree in respect of valid credit cards Y_1 to Y_8) of the kind disclosed in Fig. 1. It is recalled that the directory is an un-trusted party and therefore the merchant wants to verify that Y_5 indeed appears in the tree.

[0041] The collision intractable function h() serves for authenticating item(s) (credit cards) Y_1 to Y_8 sorted according to credit card number, e.g. in ascending order. Thus, in order to authenticate Y_5, it is sufficient for the directory to transmit to the merchant tree leaf and node values Y_5, H(6,6,Y), H(7,8,Y) and H(1,4,Y), assuming that the root value H(1,8,Y) was previously authenticated, e.g. using a digital signature. Of course, additional tree values may be transmitted but as will be appreciated from the description below transmitting additional tree values is absolutely redundant.

[0042] Thus, in order to authenticate Y_5, the merchant (knowing *a priori* H()) calculates the authentication path, namely, H(5,5,Y) (on the basis of Y_5) and on the basis of H(5,5,Y) and the so received H(6,6,Y), the merchant calculates H(5,6,Y). The latter, along with so received H(7,8,Y) give rise to H(5,8,Y). The latter along with the so received H(1,4,Y) give rise to H(1,8,Y) which is subject to PKI technique (e.g. applying the public key *n*), and the result is compared to the previously authenticated H(1,8,Y) value and in the case of match, it is assured that the item Y_5 belongs to the list of valid credit cards.

[0043] The advantage of the authentication tree is, of course, that only few tree node values were transmitted to the user which could nevertheless authenticate the item Y_5 of interest As will be explained in below, the specified description for authenticating items in respect of prior art authentication tree applies also to the search authenticated tree of the invention.

[0044] The major drawback of the authentication tree of Fig. 1 arises when the latter is subject to modify transaction, e.g. when new credit card is added to the list at the CA. Suppose that new item $Y_{4'}$ such that $Y_4 < Y_{4'} < Y_5$ is added. The resulting authentication tree (not shown) will necessitate extensive update of most of the nodes in the tree and undue transmission overhead of the updated information, which is obviously undesired, particularly when bearing in mind that the rate of updating the CA with new items is as a rule quiet high.

[0045] The advantages and disadvantages equally apply when considering a certificate revocation list (CRL) which holds the invalidated or revoced items (e.g. invalid credit cards

[0046] Considering now an exemplary search authenticated tree according to one embodiment of the invention utilizing e.g. a 2-3 search tree with a CRL (e.g. a list of revoked credit cards held at the leaves, See Fig. 2A-B.)

[0047] In this connection it should be noted that the invention is, by no means, bound to the actual realization of the search tree and any known technique that is utilized to this end is applicable, all as required and appropriate depending upon the particular application. Thus, by way of non limiting example, any manner of holding the items in the leaves is applicable, e.g. as records, link list, tree, of blocks (in the case of long item) etc. This statement is likewise valid to the authentication tree.

[0048] Thus, by this particular embodiment, a 2-3 tree is maintained with leaves corresponding to the revoked certificates' serial numbers (c1-c7) in increasing order. (In a 2-3 tree every interior node has two or three children and the paths from root to leaves have the same length). Testing membership and modifying, i.e. inserting, deleting or updating a single element are done in logarithmic time, where the modification affects only the nodes on the modification path. For a detailed presentation of 2-3 trees see [1, pp.169-180].) The property of 2-3 trees is that test and modification involve only changes to nodes on a search path, i.e. every change is local and the number of affected paths is small.

[0049] The tree may be created either by inserting the serial numbers of the revoked certificates one by one into an initially empty 2-3 tree, or, by sorting the list of serial numbers and building a degree 2 tree with leaves corresponding to the serial numbers in the sorted list (because the communication complexity is minimal when the tree is of degree 2).

[0050] Every tree node is assigned a value according to the following procedure:

- Each leaf stores a revoked certificate serial number as its value.
- The value of an internal node is computed by applying the cryptographic hash function H() to the values of its children and to at least the dynamic search values of the internal node (which encompasses also link, whenever applicable). Whilst it is not obligatory, the cryptographic one way hash function H() may also be applied to information, other than the dynamic search values that are associated with the node, e.g. information relevant for balancing the tree etc.

[0051] Unlike the collision intractable function, applying the universal one way hash function to the internal nodes in the manner specified, necessitates utilization of unique function for each node. For the latter case, it is required to authenticate in addition to the above referred to values of the children and the dynamic search values of the internal node, also the unique value of the function that is associated to the internal node.

[0052] There follows now a description that pertains to modifying the search authenticated tree according to one embodiment of the invention.

[0053] Thus, in order to delete an item, a conventional 2-3 delete item step is executed, namely:

1. Delete each expired certificate serial number from the 2-3 tree, updating the values of the nodes on the deletion path.

[0054] Likewise, in order to insert an item, a conventional 2-3 insert item step is executed, namely:

2. Insert each newly revoked certificate serial number into the tree, updating the values of the nodes on the insertion path.

[0055] During tree update, some new nodes may be created or some nodes may be deleted due to the balancing of the 2-3 tree. These nodes occur only on the search path for an inserted/deleted node (hereinafter: modified node).

[0056] The certification authority authenticates the tree by authenticating the root and to this end, only the search path that is induced by the modified nodes should be computed.

[0057] For a simpler implementation of thesearch authenticated tree, other trees, e.g. random treaps [2], may be used instead of 2-3 trees. Treaps are binary trees whose nodes are associated with (key, priority) pairs. The tree is a binary search tree with respect to node keys (i.e. for every node the keys in its left (resp. right) subtrees are small (resp. greater) than its key); and a heap with respect to node priorities (i.e. for every node its priority s higher than its descendents' priorities). Every finite set of (key, priority) pairs has a unique representation as a treap. In *random treaps*, priorities are drawn at random from a large enough ordered set (thus, they are assumed to be distinct).

[0058] Seidel and Aragon [2] present simple algorithms for membership queries, insert and delete operations with expected time complexity logarithmic in the size of the set S stored in the treap. Random treaps may be easily converted into authenticated search data structures similarly to 2-3 trees. The communication costs of these schmes is similar since the expected depth of a random treap is similar to its 2-3 tree counterpart.

- The main advantage of random treaps is that their implementation is much more simple than the implementation of 2-3 trees.
- A drawback of using random treaps is that their performance is not guaranteed in worst case, e.g. some users may (with low probability) get long authentication paths.
- Another drawback is that a stronger assumption is needed with respect to the directory. The analysis of random treaps is based on the fact that the adversary does not know the exact representation of a treap. A dishonest directory with ability to change the status of certificates may increase the computational work and communication costs of the system.

[0059] The operation of a system of the invention will be exemplified in one non-limiting sequence of operation which refers to an embodiment of the invention as depicted in Fig. 3.
Generally speaking there is provided a method in a CA, directory, user scheme, including the steps of:

(a) the user providing to a directory a list of at least one item for authenticating membership or non membership of said at least one item in a set;
(b) the directory computing and transmitting to a user the authentication path(s) as induced by said at least one item; the directory further transmitting said authenticated root; and
(c) the user verifying said items.

Still further there is provided a method in a CA directory user scheme comprising the steps of:
the CA executing:

(i) updating said search tree so as to obtain updated nodes;
(ii) computing an authentication path as induced by said updated nodes; and
(iii) authenticating at least said root modified node by a digital signature;
(iv) transmitting modified parameters to said directory; the directory executing:

(i) applying said modification parameters, so as to obtain authenticated directory root value;
(ii) verifying that the authenticated CA root value matched the authenticated directory value.

[0060] A specific description of the general aspect above will now be described:

CA Operations

[0061]

- **Creating certificates:** The CA produces a certificate by authenticating a message containing certificate data (e. g. user name and public key), certificate serial number and expiration date.
- **Initialization**: The CA creates the 2-3 tree, as above, for the set of initially revoked certificates. It computes and stores the values of all the tree nodes and sends to the directory the (sorted) list of revoked certificate serial numbers along with a signed message containing the tree root value, the tree height and a time stamp.
- **Updating:** The CA updates the tree by inserting/deleting certificates from it. After each insertion/deletion, all induced nodes are updated and the authenticated path is calculated accordingly. To update the directory, the CA sends a modification parameters. The latter may be for example the list of induced nodes, the list of the transactions. In fact modification parameter encompass any kind of information that enables the directory to update the tree at the directory end. Of course, authenticating the root encompasses of course the new root value but may likewise include other authenticated information e.g. tree height and time stamp.

Directory operations:

[0062]

- **Initialization:** Upon receiving the initial revoked certificates list, the directory computes by itself the whole 2-3 tree, checks the root value, tree height and time stamp, and verifies the CA's signature on these values.
- **Response to CA's update**: The directory updates the tree according to the modified parameters received from the CA. This results in recomputed path and authenticated directory root. Having done so it checks verifies the so obtained root value vis-a-vis the received authenticated root value as received from the CA in order to determine match, in which case the procedure terminates successfully. By this particular embodiment the root value, tree height and time stamp, all have to match (the time, of course, within reasonable interval).
- **Response to user's queries:** To answer a user query, the directory supplies the user with the authenticated root value, tree height and time stamp.

1. If the queried certificate is revoked, for each node in the path from the root to the leaf corresponding to the queried certificate, the directory supplies the user its value and its children values.
2. If the queried certificate is not revoked (not in the list), the directory supplies the user the paths to two neighbouring leaves l_1, l_2 such that the value of l_1 (resp. l_2) is smaller (resp. larger) than the queried serial number.

[0063] Note that to reduce the communication costs, the directory need not send the node values on the path from root, but only those which are required for the user to compute the entire search path. The latter was exemplified in reference to Fig. 1 where, as recalled, only Y5, H(6,6,Y), H(7,8,Y) and H(1,4,Y) were required in order to authenticate the search path of Y5.

User Operations:

[0064] The user first verifies the CA's signature on the certificate and checks the certificate expiration date. Then, the user issues a query by sending the directory the certificate serial number s. Upon receiving the directory's answer to a query, the user verifies the CA's signature on the root value, tree height and time stamp.

1. If the directory claims the queried certificate is revoked, the user checks the leaf to root path supplied by the directory by applying the hash function *h*.
2. If the directory claims the queried certificate is not revoked, the user checks the two paths supplied by the directory and checks that they lead to two adjacent leaves in the 2-3 tree, with values l_1, l_2. The user checks that $l_1 < s < l_2$. As shown in Fig. 2B for authenticating x (i.e. claiming that x does not belong to the revocation list) the search path that authenticate the adjacent members *x1* and *x2* are transmitted, where $x1 < x < x2$.

[0065] In the above scheme, the communication costs of verifying that a certificate was not revoked may be twice the communication costs of verifying that a certificate is in the list. To overcome this, the tree may be built such that every node corresponds to two consecutive serial number thus having to send only one path in either case. Since the number of bits needed for holding the value of a tree node, i.e. the hash function security parameter (l_{hash} in the notation below) is more than twice the bits needed for holding a certificate serial number, this does not influence the tree size. In this connection it is recalled that *certificate* or *item* embrace, amongst the other, range of values.

[0066] Attention is now drawn to Fig. 4 illustrating a system configuration according to another embodiment of the invention. Thus, some protocols avoid the need for a revocation system by using short-term certificates. (e.g. micro-payments protocols when a certificate owner may cause a limited damage [13]). These certificates are issued daily and expire at the end of the day of issue. Actually, even shorter periods are desired and the main limit is due to the increase in the certification authority computation (certificates for all users have to be computed daily) and communication (certificates should be sent to their owners) short-term certificates cause.

[0067] An on-line/off-line digital signature scheme (like CRS) will reduce the computation the CA has to perform, but, it will not reduce significantly the communication costs, since the CA has to send *different* messages to *different* users, making the CA a communication bottleneck. This calls for a solution where the CA performs a simple computation (say, concerning only new users and users whose certificates are not renewed) and sends a common update message to *all* users. Using this message, exactly all users with non-revoked certificates should be able to prove the validity of their certificates. To meet the latter embodiment, a simple modification to the certificate revocation scheme is proposed to yield an efficient certificate update scheme in which the CA sends the same update message to all users. In this solution there is no assumption of the existence of a directory (See Fig. 4) with information about all certificates, but of local directories that may hold the latest messages that are sent by the directory.

[0068] As before, the scheme is based on a tree of revoked certificates (or, otherwise, valid certificates) created by the certification authority presented above. Since there is no way to extract certificates from a directory, every user gets an initial certificate that may be updated using the CA's messages. Specifically, the CA augments every issued certificate with the path proving its validity, this is the only part of the certificate that is updated periodically.

[0069] To update all certificates simultaneously, the CA updates its copy of the tree, and publishes the tree paths that where changed since the previous update (constituting one, non limiting form of induced sub-tree). (see Fig. 5A). Every user holding a non-revoked certificate intersects its self path with the induced tree preferably by locating the lowest node, *v*, on a path that coincides with the self path, and updates his path by copying the new node values from v up to the root). All users holding a revoked certificate can not update their path, unless they crack the one way characteristics of the function *hl*. As shown in Fig. 5B the user bring into coincidence the self path with the induced sub tree and seeks for the lower most discrepancy node (desgnated by dot 100 in Fig. 5A). What remains to be done is to update the nodes from the so detected node to the root and to authenticate the root and verify it vis-avis the authenticated root value transmitted from the CA. The latter procedure is obviously very cost effective in terms of the computation overhead that is posed on each user.

[0070] Since the CA communication is reduced, one may use this update scheme for, say, updating certificates once every hour. This may cause some users to lag in updating their certificates, and the local directories should save several latest update messages (e.g. using conventional proxy servers), and some aggregate updates (combining update messages of a day) enabling users that lag several days to update their certificates.

[0071] The latter specific description is defined more generally as follows, a method according to Claim 6, in a CA

user scheme comprising:
the CA executing:

(i) updating said search tree so as to obtain updated nodes;
(ii) computing an authentication path as induced by said updated nodes; and
(iii) authenticating at least said root modified node by a digital signature;
(iv) transmitting induced sub-tree to said user;

the user executing:

(i) intersecting said induced sub-tree with user self path and obtaining user authenticated root value;
(ii) verifying that the authenticated CA root value matched the authenticated user value.

[0072] Those versed in the art will readily appreciate that the realization of the embodiments of Figs. 3 and 4 are not bound to any specific hardware and/or software architecture. Thus, by way of non limiting example, the CA, directory and user may be interlinked by any available communication network, e.g. the Internet. By way of another non limiting example each of the specified constituents may be implemented on e.g. conventional P.C. computer, mainframe computer or network of computers, all as required and appropriate, depending upon the particular application.

Evaluation

[0073] In the following, the communication costs of CRL, CRS and one, non limiting, embodiment of a system/method of the invention are compared. Basing on this analysis, there is shown that the proposed system is more robust to changes in parameters, and allows higher update rates than the other.

[0074] Other advantages of the proposed scheme are.

- The CA has to keep a smaller secret than in CRS.
- Since CA-to-directory communication is low, the CA may communicate with the directory using a slow communication line secured against breaking into the CA's computer (the system security is based on the ability to protect the CA's secrets).
- In the case of a 2-3 tree, there is never a need to re-compute the entire tree to update it. This allows higher update rates than CRT.
- Another consequence of the low CA-to-directory communication is that a CA may update many directories, avoiding bottlenecks in the communication network.

Communication Costs

[0075] The parameters we consider are:

- n - Estimated total number of certificates ($n = 3,000,000$).
- k - Estimated average number of certificates handled by a CA ($k = 30,000$)
- p - Estimated fraction of certificates that will be revoked prior to their expiration ($p = 0.1$). (It is assumed that certificates are issued for one year, thus, the number of certificates revoked daily is $\frac{n.p.l_{sn}}{365}$: 365
- q - Estimated number of certificate status queries issued per day
- ($q = 3,000,000$).
- T- Number of updates per day ($T = 1$).
- l_{sn} - Number of bits needed to hold a certificate serial number ($l_{sn} = 20$).
- l_{stat} - Number of bits needed to hold the certificate revocation status numbers Y_{365-i} and N_0 ($l_{stat} = 100$).
- l_{sig}- Length of signature ($l_{sig} = 1,000$).
- l_{hash} - Security parameter for the hash function ($l_{hash} = 128$).

[0076] Values for n,k,p,q,T,l_{sn}, l_{stat} are taken from Micali [18], l_{sig} and l_{hash} are specific to our scheme.

CRL Costs

[0077]

- The CRL daily update cost is $T.n.p.=l_{sn}$ since each CA sends the whole CRL to the directory in each update. An

alternative update procedure where the CA sends to the directory only a difference list (which serial numbers to add/remove from the previous CRL) costs: $\frac{n.p.l_{sn}}{365}$

- The CRL daily query costs is $q.p.kl_{sn}$ since for every query the directory sends the whole CRL to the querying user.

CRL Cost

[0078]

- The CRS daily update cost is $T.n.(l_{sn} + l_{stat})$ since for every certificate the CA sends l_{stat} bits of certificate revocation status.
- The CRS daily query cost is $l_{stat}.q$.

The proposed scheme

[0079] To update the directory, the CA sends the difference lists of total daily length of $\frac{n.p.l_{sn}}{365} + T.l_{sig}$

- To answer a user's query, the directory sends up to 2 . $\log_2(p \cdot k)$ numbers, each l_{hash} bits long, totaling $2.q.l_{hash}.\log_2(p \cdot k)$ bits.

[0080] The following table shows the estimated daily communications costs (in bits) according to the three schemes.

	CRL Costs	CRS Costs	Proposed Scheme
Daily update (CA-directory)	$6\text{-}10^6$	$3.6\text{-}10^8$	$1.7\text{-}10^4$
Daily queries (Directory-users)	$1.8\text{-}10^{11}$	$3\text{-}10^8$	$7\text{-}10^9$

[0081] As shown in the table, the proposed scheme costs are lower than CRL costs both in CA-to-directory and in directory-to-users communication. The CA-to-directory costs are much lower than the corresponding CRS costs but, the directory-to-user (and thus the over all) communication costs are increased. Note that in practice, due to communication overheads, the difference between CRS and the proposed method in Directory-to-users communication may be insignificant.

[0082] The proposed scheme is more robust to changes in parameters than CRL and CRS. Since these are bound to change in time or due to the specific needs of different implementations, it is important to have a system that is robust to such changes.

[0083] Changes will occur mainly in the total number of certificates (n) and the update rate (T). In the proposed method, changes in n are moderated by a factor of p. Changes in T are moderated by the fact that the update communication costs are not proportional to nT but to T. Figure 8 shows how the CA-to-directory update communication costs of the three methods depend on the update rate (all other parameters are held constant). The update communication costs limit CRS to about one update a day (Another factor that limits the update rate is the amount of computation needed by a user in order to verify that a certificate was not revoked). The proposed scheme is much more robust, even allowing once per hour updates.

[0084] The present invention has been described with a certain degree of particularity, but it should be understood that various modifications and alterations may be made Without departing from the scope or spirit of the invention as defined by the following claims:

Claims

1. A memory containing an authenticated search tree that serves for authenticating membership or non membership of items in a set; the authenticated search tree, comprising:

   a search tree having nodes and leaves and having associated therewith a search scheme; the nodes including

dynamic search values and the leaves including items of said set; the nodes are associated, each, with a cryptographic hash function value that is produced by applying a cryptographic hash function to at least: (I) the cryptographic hash values of the children nodes and (II) the dynamic search value of said node; at least the root node of said authenticated search tree is authenticated by a digital signature.

2. An authenticated search tree according to claim 1 wherein said cryptographic hash function being of the universal one way function type, and wherein said cryptographic one way function is further applied to a universal one way function that is unique to each internal node.

3. A search authenticated tree of Claim 1, wherein said search tree being Btree.

4. A search authenticated tree of Claim 1, wherein said search tree being 2-3. tree.

5. A method for authenticating membership or non membership of items in a set; comprising:

(i) providing an authenticated search tree as defined in Claim 1;
(ii) authenticating at least one item of said set by computing the authentication path as induced by said at least one item and the root.

6. A method for updating at least one item of a set in an authenticated search tree, comprising:

(i) providing a search authenticated tree as defined in Claim 1;
(ii) updating said search tree so as to obtain updated nodes;
(iii) computing an authentication path as induced by said updated nodes; and
(iv) authenticating at least said root modified node by a digital signature.

7. A method according to Claim 5, in a CA, directory, user scheme, wherein said step (ii), includes:

(a) the user providing to a directory a list of at least one item for authenticating membership or non membership of said at least one item in a set;
(b) the directory computing and transmitting to a user the authentication path(s) as induced by said at least one item; the directory further transmitting said authenticated root; and
(c) the user verifying said items.

8. A method according to Claim 6, in a CA directory user scheme comprising the steps of:
the CA executing:

(i) updating said search tree so as to obtain updated nodes;
(ii) computing an authentication path as induced by said updated nodes; and
(iii) authenticating at least said root modified node by a digital signature;
(iv) transmitting modified parameters to said directory;
the directory executing:

(i) applying said modification parameters, so as to obtain authenticated directory root value;
(ii) verifying that the authenticated CA root value matched the authenticated directory value.

9. A method according to Claim 6, in a CA user scheme comprising:
the CA executing:

(i) updating said search tree so as to obtain updated nodes;
(ii) computing an authentication path as induced by said updated nodes; and
(iii) authenticating at least said root modified node by a digital signature;
(v) transmitting induced sub-tree to said user;

the user executing:

(iii) intersecting said induced sub-tree with user self path and obtaining user authenticated root value;
(iv) verifying that the authenticated CA root value matched the authenticated user value.

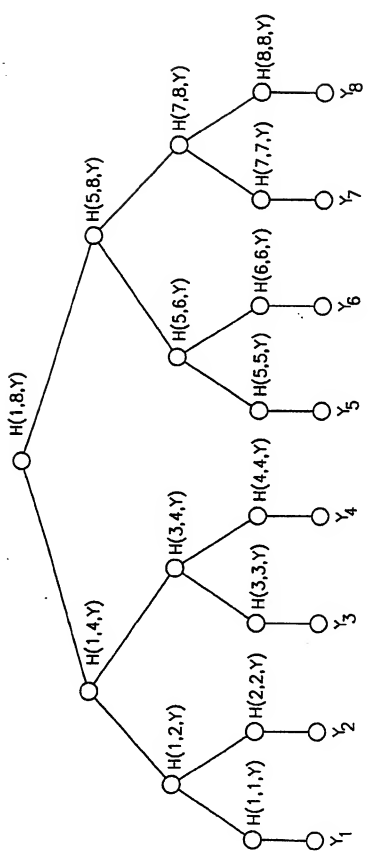
H(1,8,Y)
H(1,4,Y)
H(5,8,Y)
H(1,2,Y)
H(3,4,Y)
H(5,6,Y)
H(7,8,Y)
H(1,1,Y)
H(2,2,Y)
H(3,3,Y)
H(4,4,Y)
H(5,5,Y)
H(6,6,Y)
H(7,7,Y)
H(8,8,Y)
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8

FIG. 1

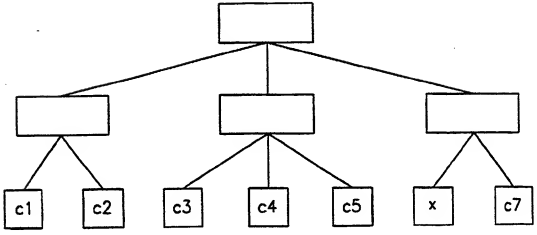

FIG.2A

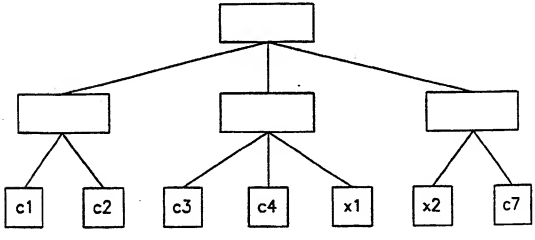

FIG.2B

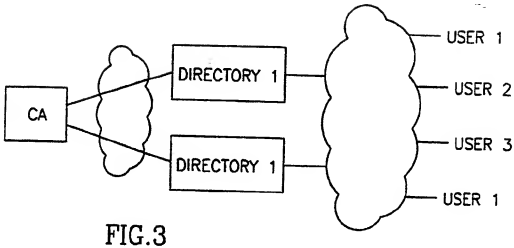

FIG.3

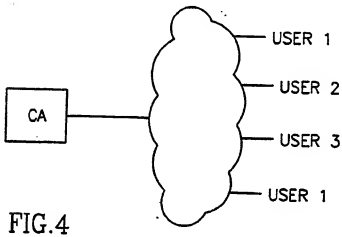

FIG.4

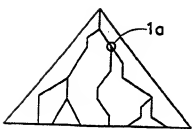

FIG.5A

FIG.5B

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 932 109 A3

(12) **EUROPEAN PATENT APPLICATION**

(88) Date of publication A3:
18.06.2003 Bulletin 2003/25

(51) Int Cl.7: **G06F 17/30**, H04L 9/32

(43) Date of publication A2:
28.07.1999 Bulletin 1999/30

(21) Application number: **99400130.3**

(22) Date of filing: **21.01.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **22.01.1998 US 10571**

(71) Applicant: **YEDA RESEARCH & DEVELOPMENT COMPANY, LTD.**
76100 Rehovot (IL)

(72) Inventors:
- **Naor, Moni**
  Tel Aviv 69122 (IL)
- **Nissim, Yaacov**
  Ramat-Gan 52525 (IL)

(74) Representative: **Moutard, Pascal Jean et al**
Cabinet Beau de Loménie
158, rue de l'Université
75340 Paris Cedex 07 (FR)

(54) A method for authentification item

(57) A memory containing an authenticated search tree that serves for authenticating membership or non membership of items in a set. The authenticated search tree including a search tree having nodes and leaves and being associated with a search scheme. The nodes including dynamic search values and the leaves including items of the set The nodes are associated, each, with a cryptographic hash function value that is produced by applying a cryptographic hash function to the cryptographic hash values of the children nodes and to the dynamic search value of the node. The root node of the authenticated search tree is authenticated by a digital signature.

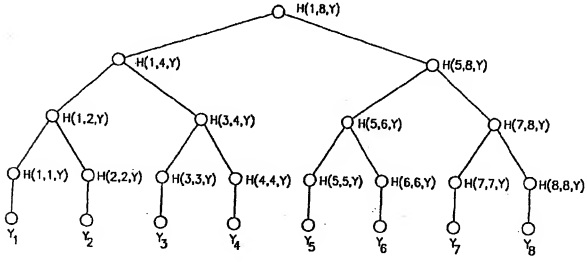

FIG.1

EP 0 932 109 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 40 0130

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
D,A	US 4 309 569 A (MERKLE RALPH C) 5 January 1982 (1982-01-05) * the whole document * ---	1	G06F17/30 H04L9/32
D,A	AHO ET AL.: "DATA STRUCTURES AND ALGORITHMS" 1982 , ADDISON-WESLEY , READING (US) XP002238504 * page 169 - page 174 * ---	1	
P,X	US 5 826 254 A (KAHN CLIFFORD EARL) 20 October 1998 (1998-10-20) * column 3, line 59 - column 4, line 13 * ---	1	
P,X	NAOR M ET AL: "Certificate revocation and certificate update" PROCEEDINGS OF THE SEVENTH USENIX SECURITY SYMPOSIUM, PROCEEDINGS OF THE SEVENTH USENIX SECURITY SYMPOSIUM, SAN ANTONIO, TX, USA, 26-29 JAN. 1998, pages 217-228, XP002238503 1998, Berkeley, CA, USA, USENIX Assoc, USA ISBN: 1-880446-92-8 * the whole document * ---	1-9	
A	WO 97 43842 A (INTEGRIS SECURITY INC) 20 November 1997 (1997-11-20) * abstract * * page 16, line 9 - page 17, line 7 * * page 25, line 18 - page 26, line 5 * -----	1	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)
H04L
G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	16 April 2003	Holper, G

CATEGORY OF CITED DOCUMENTS

X: particularly relevant if taken alone
Y: particularly relevant if combined with another document of the same category
A: technological background
O: non-written disclosure
P: intermediate document

T: theory or principle underlying the invention
E: earlier patent document, but published on, or after the filing date
D: document cited in the application
L: document cited for other reasons

&: member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 40 0130

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

16-04-2003

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 4309569	A	05-01-1982	NONE		
US 5826254	A	20-10-1998	NONE		
WO 9743842	A	20-11-1997	US	5903651 A	11-05-1999
			AU	3124997 A	05-12-1997
			GB	2330504 A	21-04-1999
			WO	9743842 A1	20-11-1997
			US	6532540 B1	11-03-2003
			US	2002188843 A1	12-12-2002
			US	6442689 B1	27-08-2002